

PATENT ABSTRACTS OF JAPAN

13

(11)Publication number : 05-342045
 (43)Date of publication of application : 24.12.1993

(51)Int.Cl. G06F 11/22
 G06F 9/455

(21)Application number : 04-149715 (71)Applicant : FUJIKURA LTD
 (22)Date of filing : 09.06.1992 (72)Inventor : SUGIYAMA SHIGEKI

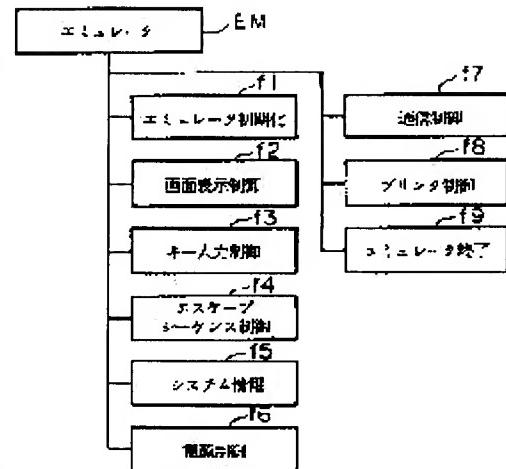
(54) EMULATOR

(57)Abstract:

PURPOSE: To emulate the touch panel input of a handy terminal on a developing machine by providing an emulating means.

CONSTITUTION: As a library formed from plural routines called by a program, an emulating means (emulator) EM is provided for the respective function routines defined by the same argument as the program to respectively emulate respective functions corresponding to the hardware configuration on the other computer side.

Namely, an input form regulated at the program prepared for one computer is emulated by the emulating means EM while using an input means provided at the other computer. As a result, the program prepared for one computer can be operated on the other computer between the computers for which the hardware configuration is made different each other, and the touch panel input operation performed on one computer side can be emulated by the mouse input of the other computer.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平5-342045

(43)公開日 平成5年(1993)12月24日

(51)Int.Cl.⁵
G 0 6 F 11/22
9/455

識別記号 庁内整理番号
3 6 0 B 8323-5B

F I

技術表示箇所

9292-5B

G 0 6 F 9/ 44

3 1 0 A

審査請求 未請求 請求項の数1(全12頁)

(21)出願番号 特願平4-149715

(22)出願日 平成4年(1992)6月9日

(71)出願人 000005186

株式会社フジクラ

東京都江東区木場1丁目5番1号

(72)発明者 杉山 茂樹

千葉県佐倉市六崎1440番地 藤倉電線株式

会社佐倉工場内

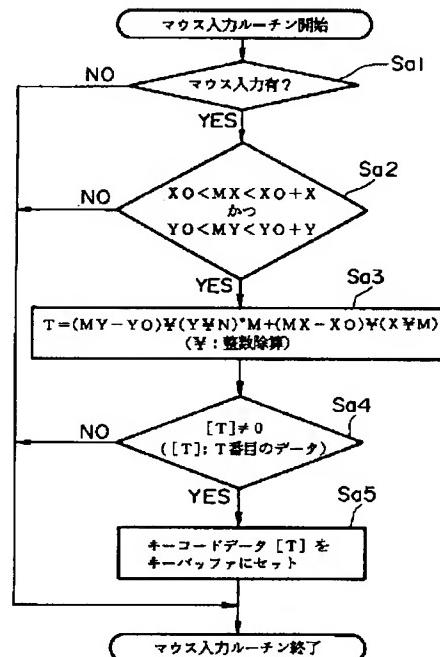
(74)代理人 弁理士 志賀 正武

(54)【発明の名称】 エミュレータ

(57)【要約】

【目的】 ハンディターミナル1のタッチパネル入力を開発マシン上で模倣できるエミュレータを実現する。

【作用】 ハンディターミナル1用に作成したプログラムと同一の引数で定義された各関数ルーチンから構成されるエミュレータEMが、それぞれパソコンコンピュータ側のハードウェア構成に対応した各機能をエミュレートすると共に、このエミュレートされるハンディターミナル1のタッチパネル入力操作を、パソコンコンピュータにおけるマウス入力で模倣する。



【特許請求の範囲】

【請求項1】互いにハードウェア構成が異なるコンピュータの内、いずれか一方のコンピュータ用に作成したプログラムを、他方のコンピュータ上で動作可能にするエミュレータにおいて、

前記プログラムがコールする複数のルーチンから形成されるるライブラリであって、前記プログラムと同一の引数で定義された各関数ルーチンが、それぞれ前記他方のコンピュータ側のハードウェア構成に対応した各機能を模倣する模倣手段を具備し、

前記模倣手段は、前記プログラムで規定された前記一方のコンピュータ側の入力形態を、前記他方のコンピュータが備える入力手段を用いて模倣することを特徴とするエミュレータ。

【発明の詳細な説明】

【産業上の利用分野】

【0001】この発明は、例えば、「ハンディターミナル」と呼ばれる携帯可能なコンピュータ用のアプリケーションプログラム開発用いて好適なエミュレータに関する。

【0002】

【従来の技術】近年、「ハンディターミナル」と呼ばれる携帯可能なコンピュータが各種実用化されている。この種のコンピュータは、ポータブルに構成されており、携行先で種々のデータ処理が可能になるため、様々な分野で使用されている。図8は、このようなハンディターミナル1の一構成例を示すブロック図である。

【0003】この図において、1aはハンディターミナル1の各部を制御するCPUである。1bは、CPU1aの基本的な動作を制御するオペレーティングシステムプログラム（以下、これをOSと略す）が記憶されるROMである。1cは対応業務用のアプリケーションプログラムがロードされると共に、該プログラムのワークエリアとして各種レジスタ値が一時記憶されるRAMである。1dは、LCD（液晶表示素子）等から構成される表示回路であり、内部バスを介してCPU1aから供給される各種データを表示する。1eはLCD（液晶表示素子）上に設けられた透明タッチパネル、あるいは本体パネルに配置されるテンキーやファンクションキー等から構成される操作子であり、それぞれ各操作に応じた操作子信号を発生する。

【0004】1fは本体から着脱自在に構成されるメモリカードである。このメモリカード1fには、上述したアプリケーションプログラム、あるいは当該プログラムによって参照される各種データが記憶される。なお、このメモリカード1fが本体に装着されていない場合には、上述したRAM1cの一部をRAMディスクとして使用し、ここにアプリケーションプログラムや各種データを記憶することも可能である。また、メモリカード1fに記憶されるデータは、図示されていない上位コンピ

ュータからダウンロードされるものである。1gは、例えば、モ뎀等から構成され、シリアルデータ通信を制御する通信制御回路である。このような構成によるハンディターミナル1は、周知のコンピュータと同様に動作する。つまり、電源投入後にOSが立ち上がると共に、このOS上で対応業務用のアプリケーションプログラムがデータ処理を実行する。

【0005】ところで、ハンディターミナル1で動作するアプリケーションプログラムは、「C言語」と呼ばれる構造化プログラミング言語で記述される場合が多い。

図9は、「C言語」によるプログラム開発手順を示す図である。この図に示すように、プログラム開発は、コーディング、コンパイルおよびリンクの各作業からなり、実機デバッグを経て当該プログラムの動作が検証される。なお、このコーディング、コンパイルおよびリンクの各作業は、通常、開発マシン（例えば、パーソナルコンピュータ）上で行われる。

【0006】コーディングにおいては、予め定められたシステム仕様に基づき、ソースプログラムをC言語で記述する。このソースプログラムは、ソースファイルsfとして開発マシン上に登録される。ソースファイルsfは、コンパイラCCの入力ファイルとなる。コンパイラCCでは、C言語で記述されたソースプログラムを語彙解析、構造解析および意味解析し、この結果をリロケータブルな中間コードで記述されたオブジェクトプログラムに変換する。このオブジェクトプログラムは、開発マシン上でオブジェクトファイルofとして登録される。

【0007】リンクLKでは、オブジェクトファイルofに対し、標準関数ライブラリSLと、ターミナル専用

30 関数ライブラリTLとを結合させ、マシン語で記述された実行ファイルEfを生成する。ここで、標準関数ライブラリSLは、C言語において定義される各種の制御関数ルーチンプログラムから構成されている。また、ターミナル専用関数ライブラリTLは、ハンディターミナル1のハードウェア環境で動作するように定義された専用プログラム群から構成されるものである。

【0008】すなわち、リンクLKでは、オブジェクトプログラムにおいてコールされる種々のルーチンプログラムや専用プログラムが上述した各ライブラリSL、TLから引用され、これらをオブジェクトプログラムに結合させる。これにより、リロケータブルな中間コードで記述されたプログラムが実行形式に変換され、実行ファイルEfとして生成される。このようにして生成される実行形式のアプリケーションプログラムは、絶対アドレス上に展開可能な形態となる。

【0009】次に、実行ファイルEfは、例えば、開発マシンから前述したメモリカード1fに書き込まれ、該メモリカード1fを介してCPU1aにダウンロードDLされる。これにより、ハンディターミナル1上で実機デバッグDG1が施される。実機デバッグDG1におい

ては、予め策定された検査項目に従ってアプリケーションプログラムの動作を検証する。この実機デバッグDG1でバグが露見した場合には、ソースプログラムの対応箇所を修正する。そして、修正されたソースプログラムは、再びコンパイル、リンク作業を経て実行ファイル化されて実機デバッグDG1が繰り返される。

【0010】

【発明が解決しようとする課題】さて、上述したプログラム開発手順においては、ハンディターミナル1上でバグが露見する度毎に、当該バグに対応するソースプログラム部分を修正し、これを再度実行ファイル化してハンディターミナル1にダウンロードしなければならない。このため、実機デバッグDG1には多大な工数が費やされ、結果的に開発コスト上昇を招致するという弊害がある。

【0011】そこで、こうした弊害を解決するには、上述したコーディング、コンパイル、リンクおよびデバッグからなる一連の作業を全て開発マシン上で行い、かつ、デバッグ作業時には、特に、開発マシン上でハンディターミナル1の動作状態を全て把握できる形態になることが要求される。これを換言すれば、開発マシン上でアプリケーションプログラムを実行し、ハンディターミナル1の動作をエミュレートできれば、上述した欠点が解消され、効率良いプログラム開発が可能になる訳である。

【0012】ところが、ハンディターミナル1と開発マシンとは、殆どの場合、ハードウェア環境が全く異なるため、ハンディターミナル1用に作成されたアプリケーションプログラムを開発マシン上で動作させることができない。特に、ハンディターミナル1においては、前述したように、タッチパネルを操作することで各種入力操作がなされるように構成されている。したがって、開発マシンとなるパソコンコンピュータにおいて、タッチパネル入力を実現するためには、該コンピュータのディスプレイに専用のタッチパネルを新たに設ける必要になる。

【0013】しかしながら、新たに専用のタッチパネルを購入するというのも極めて不経済である。また、通常のディスプレイにタッチパネルを装着すると、画面とタッチパネル面との間に隙間ができ、これにより視差が生じる。すなわち、画面に表示された領域と、これに対応するタチパネルのキー領域とがずれて見えてしまい、入力操作がし難くなる。加えて、パソコンコンピュータ用のタッチパネルでは、実際のハンディターミナル1のように、細かいキー領域に分割することも期待できない。

【0014】結局、現状においては、開発マシンにタッチパネルを設けることなく、ハンディターミナル1のタッチパネル入力やその他の動作をエミュレートし、極めて効率良いプログラム開発環境を実現することが急務と

されている。この発明は上述した事情に鑑みてなされたもので、ハンディターミナル1のタッチパネル入力を開発マシン上で模倣できるエミュレータを提供することを目的としている。

【0015】

【課題を解決するための手段】この発明は、互いにハードウェア構成が異なるコンピュータの内、いずれか一方のコンピュータ用に作成したプログラムを、他方のコンピュータ上で動作可能にするエミュレータにおいて、前記プログラムがコールする複数のルーチンから形成されるライブラリであって、前記プログラムと同一の引数で定義された各関数ルーチンが、それぞれ前記他方のコンピュータ側のハードウェア構成に対応した各機能を模倣する模倣手段を具備し、前記模倣手段は、前記プログラムで規定された前記一方のコンピュータ側の入力形態を、前記他方のコンピュータ側が備える入力手段を用いて模倣することを特徴としている。

【0016】

【作用】この発明によれば、模倣手段は、一方のコンピュータ用に作成したプログラムにおいて規定された入力形態を、他方のコンピュータが備える入力手段を用いて模倣する。この結果、互いにハードウェア構成が異なるコンピュータ間において、一方のコンピュータ用に作成されたプログラムを他方のコンピュータ上で動作可能とし、一方のコンピュータ側でなされるタッチパネル入力操作を、他方のコンピュータのマウス入力で模倣する。

【0017】

【実施例】以下、図面を参照してこの発明の実施例について説明する。図1はこの発明の一実施例を適用したプログラム開発手順の概要を示す図である。この図において、図9に示す各部と共に通する部分には、同一の符号を付し、その説明を省略する。図1に示す手順が図9に示した従来例と異なる点は、ターミナル専用関数ライブラリTL（図9参照）を後述するエミュレータEMに置き換え、これにより、開発マシン（パソコンコンピュータ）上でハンディターミナル1用に作成されたアプリケーションプログラムのデバッグDG2を行うようにした点にある。すなわち、この実施例が意図するところは、エミュレータEMを用いたことにより、従来必要とされていた実行ファイルEfのダウンロードDLと、これに応じてなされるハンディターミナル1上の実機デバッグDG1とを省略し、かつ、開発マシン上でハンディターミナル1のタッチパネル入力操作を模倣するようにした点にある。

【0018】次に、図1のプログラム開発手順を実現するエミュレータEMの機能概要について説明する。まず、エミュレータEMは、ハンディターミナル1のハードウェア環境で動作するよう定義された各種関数を、開発マシン（例えば、パソコンコンピュータ）のハードウェア環境で動作するように定義し直したプログラム群

から構成されている。

【0019】エミュレータEMの機能は、図2に示すように、ターミナル専用関数ライブラリTL（図9参照）と同一である。つまり、このエミュレータEMでは、ターミナル専用関数ライブラリTLと同様の引数で各関数の外部仕様を規定し、かつ、各関数内部は、開発マシン（パソコンコンピュータ）上でハンディターミナル1の動作をエミュレートするよう定義し直している。

【0020】図2において、f1はエミュレータ初期化機能である。このエミュレータ初期化機能f1は、ソースプログラムの先頭に「emu_start()」なる関数が記述されている場合、エミュレータEMの初期設定を行うものである。したがって、図1に示すように、オブジェクトファイルofとエミュレータEMとをリンクさせる場合には、この「emu_start()」なる関数が必要になる。なお、この関数は、ターミナル専用関数ライブラリTLとリンクする際には、何等実行に影響されないものである。したがって、エミュレータ用に作成されたソースファイルsfは、そのままハンディターミナル1用の実行ファイルに変換可能になる。

【0021】次に、f2は画面表示制御機能である。この画面表示制御機能f2では、ハンディターミナル1用に定義された各関数、例えば、表示モードの設定、カーソル形状あるいは表示文字のアトリビュート設定等を行う関数を、それぞれ開発マシン（パソコンコンピュータ）のハードウェア環境に対応して動作させる機能である。f3はキー入力制御機能であり、ハンディターミナル1でなされるタッチパネル入力やテンキー入力を、開発マシン上におけるマウス入力やテンキー入力に置き換える機能である。f4はエスケープシーケンス制御機能であり、カーソル位置指定、全画面消去あるいはスクロールアップ／ダウンなどを指定するコード出力関数を、開発マシン（パソコンコンピュータ）のハードウェア環境に対応して動作させる機能である。

【0022】f5は、メインメモリ容量、ドライバ登録情報、外字登録情報等のシステム情報を取得して出力する関数を開発マシンに対応させた機能である。f6は主電源オン／オフ制御や、電源電圧低下状態検出などの動作を開発マシン上で表示させる機能である。f7は、ハンディターミナル1用に定義されたシリアルデータ通信制御を、開発マシン上でエミュレートさせる通信制御機能である。f8はプリント制御機能であり、ハンディターミナル1用に定義された印字フォント、改行ピッチなどを開発マシン上で制御するものである。f9は、エミュレータ終了機能であり、ソースプログラムの最後、あるいは途中に「emu_exit(n)」なる関数が記述されている場合、エミュレータEMを終了させる機能である。

【0023】次に、上記エミュレータEMを用いたプログラム開発手順と、この手順に基づき作成されたアプ

ケーションプログラムのエミュレート動作概要と、該エミュレート動作におけるマウス入力処理の動作について順次説明する。

①プログラム開発手順

アプリケーションプログラムを開発する際には、まず、コーディング段階でソースプログラムの先頭に「emu_start()」なる関数を記述し、かつ、該プログラムの最後に「emu_exit(n)」なる関数を記述しておく。そして、図1に示すように、ソースファイルsfをコンパイルccにかけ、オブジェクトファイルofを生成する。

【0024】次に、このオブジェクトファイルofに対して標準関数ライブラリSLおよびエミュレータEMをリンクさせる。これにより、ソースプログラムでコールされるハンディターミナル固有の各種関数がパソコンコンピュータ上で動作する形で実行ファイル化される。実行ファイル化されたアプリケーションプログラムは、パソコンコンピュータの主メモリ上にロードされてハンディターミナル1の動作をエミュレートする。

20 【0025】②アプリケーションプログラムのエミュレート動作

上記手順により作成されたアプリケーションプログラムが起動すると、パソコンコンピュータの処理は、図3に示すステップS1に進む。ステップS1では、プログラム先頭に定義された「emu_start()」なる関数に対応して上述したエミュレータ初期化機能f1が初期設定を行う。この初期設定とは、ハンディターミナル1の起動状態をエミュレートするものであり、例えば、バックライトのオン／オフ状態、液晶パネルのコン

30 トラスト状態あるいはスピーカ音量などを設定するものである。

【0026】こうしてパソコンコンピュータ上でハンディターミナル1の初期状態がエミュレートされると、例えば、図4に示すように、ハンディターミナル1の表示画面TDに対応した画面がパソコンコンピュータのディスプレイDSPに表示される。ここで、同図(ロ)は、実際のハンディターミナル1に電源が投入された時に表示される表示画面の一例を示す図である。一方、同図(イ)は、パソコンコンピュータに表示されるエミュレート画面の一例である。同図(イ)に示すように、ディスプレイDSPには、表示エリアE1, E2, E3が表示され、この内、表示エリアE1にはハンディターミナル1の表示画面TDと同一の内容が表示される。表示エリアE1では、後述するマウス入力処理に基づき、タッチパネル入力操作がエミュレートされる。また、表示エリアE2には、該エリアE1に対応したタッチパネル状態が表示され、さらに、表示エリアE3にはハンディターミナル1の動作状態が表示される。

【0027】次いで、このような初期設定がなされると、パソコンコンピュータの処理はステップS2に進

む。ステップS 2では、与えられたイベントに応じてアプリケーションプログラムが各関数を実行する。このステップS 2においては、後述するように、ハンディターミナル1のタッチパネル入力をマウス入力で置き換えるキー入力制御機能f 3が動作しており、これによりマウスカーソルを移動させるマウス入力処理が行われる。

【0028】したがって、実際のタッチパネルをエミュレートした表示エリアE 1上の所定位置をクリックしてマウス入力をを行うと、この入力イベントに対応したアプリケーションプログラムが起動し、該プログラム中で定義された関数が実行される。これにより、ハンディターミナル1の動作がパソコンコンピュータ上でエミュレートされる。例えば、マウスカーソルMCが図6に示す位置に置かれ、マウス入力がなされると、「データ伝送」処理が起動する。そして、アプリケーションプログラム完了の旨を表わす入力がなされると、ステップS 3の判断結果が「YES」となり、該プログラムの処理がステップS 4に進む。ステップS 4では、プログラム最後に定義された「emu_exit(n)」なる関数に基づき、エミュレート動作を終了する。

【0029】③マウス入力処理動作

次に、図5～図7を参照し、上記ステップS 2でなされるマウス入力処理について説明する。まず、上述したステップS 2において、キー入力制御機能f 3が実行されると、図5に示すマウス入力ルーチンが起動する。これにより、パソコンコンピュータの処理がステップS a 1に進む。ステップS a 1では、マウス入力、すなわち、マウスがクリックされたか否かを判断する。ここで、マウス入力が無い場合には、判断結果が「NO」となり、このルーチンを完了する。一方、マウス入力がなされた場合には、判断結果が「YES」となり、次のステップS a 2に処理が進む。ステップS a 2では、クリックされたマウス座標が「X0 < MX < X0 + X」、かつ、「Y0 < MY < Y0 + Y」であるか否かを判断する。

【0030】ここで、図7を参照し、上記マウス座標の定義について説明する。このマウス座標は、パソコンコンピュータのディスプレイDSPの表示ドットで定義されるものであって、該ディスプレイDSPの上部左端を座標原点(0, 0)、上述した表示エリアE 1の上部左端をエリア座標(X0, Y0)としている。表示エリアE 1は、原点(X0, Y0)からX方向(水平方向)へXドット分、該原点(X0, Y0)からY方向(垂直方向)へYドット分で構成された表示領域となる。このような表示領域において、X方向はM分割され、Y方向はN分割されている。これらM, Nは、キー分割数に対応し、これら分割領域は、実際の「タッチパネル」に合せたキー領域として種々定義されるようになっている。なお、(MX, MY)は、マウス入力がなされたクリック座標を表わしている。

【0031】したがって、上記ステップS a 2では、クリック座標(MX, MY)が表示エリアE 1の内部にあるか否かを判断するものである。ここで、該エリアE 1の外部でマウス入力された時、つまり、タッチパネルのキー領域以外がクリックされた時には、ここでの判断結果が「NO」となり、このルーチンを終了する。これに対し、該エリアE 1内でクリックされ、タッチパネルのキー領域に相当する位置が操作された場合には、判断結果が「YES」となり、次のステップS a 3に進む。ステップS a 3では、クリック座標(MX, MY)がタッチパネルのどのキー領域に対応するのかを算出する。すなわち、ステップS a 3に記載の式に従って、キーテーブルの読み出しアドレスTを算出する。

【0032】キーテーブルとは、実際のタッチキー分割領域と、表示エリアE 1上のマウス入力領域との対応関係を表わしたテーブルである。このキーテーブルは、予めパソコンコンピュータの内部メモリにキーテーブルファイルとして登録される。例えば、図6に示す表示例の場合、表示エリアE 1には、ハンディターミナル1のタッチキー分割領域に対応させたマウス入力領域R 1～R 7のキーデータがキーテーブルに登録される。いま、仮に、図7に示すように、マウス入力領域R 4にマウスカーソルMCが置かれ、この位置でクリックされると、該領域R 4に対応させたキーデータを読み出すためのアドレスTが算出される。

【0033】次いで、ステップS a 4では、この算出されたアドレスのキーデータ[T]が「0」であるか否かを判断する。ここで、該キーデータ[T]が「0」である場合には、キーテーブルにデータが設定されていない(キー入力無視)として判断結果が「NO」となり、このルーチンを終了する。一方、「0」でない場合には、キーテーブルにデータが設定されているので、判断結果が「YES」となり、次のステップS a 5に進む。ステップS a 5では、アドレスTに従って、上述したキーテーブルからキーコードデータを読み出し、これをキーバッファにセットしてこのルーチンを終了する。この結果、パソコンコンピュータは、キーバッファにセットされたキーデータに応じた処理を実行する。

【0034】図6に示す一例では、ハンディターミナル1のタッチキー分割領域に対応させたマウス入力領域R 1～R 7が表示エリアE 1に設けられており、これら領域R 1～R 7にマウスカーソルMCを指示してクリックすることで、実際のタッチパネル入力をエミュレートしている。すなわち、マウス入力領域R 1～R 4のいずれかをクリックすることで処理選択がなされ、マウス入力領域R 5～R 7では画面選択と終了指示とが行われる。

【0035】なお、表示エリアE 2には、上記マウス入力領域に対応するタッチキー分割状態が表示されている。この場合、タッチパネルを「3行12列」に分割しており、この内、タッチキー分割領域KR 1～KR 4を

マウス入力領域R 1～R 4に対応させ、タッチキー分割領域KR 5～KR 7をマウス入力領域R 5～R 7に対応させている。そして、各タッチキー分割領域KR 1～KR 7には、各々割り当てられたキーコードが16進数で表示される。すなわち、タッチキー分割領域KR 1～KR 4に「41」～「44」が、タッチキー分割領域KR 5～KR 7に「61」～「63」が表示される。

【0036】このように、パーソナルコンピュータ上において、キー入力制御機能f 3が実行されると、表示エリアE 1にはハンディターミナル1のタッチキー分割領域に対応させたマウス入力領域が設定され、このマウス入力領域をクリックすることで、タッチパネル入力操作がエミュレートされる。なお、表示エリアE 2には、マウス入力領域に対応させたタッチキー分割状態が表示されると共に、各タッチキー領域に割り当てられたキーコードデータが表示される。この結果、従来、直接的に動作状態を検証できなかったタッチパネル分割状態が一目瞭然になる。しかも、この場合、パーソナルコンピュータ用のタッチパネルを設けることなく、実際のタッチパネル入力をマウス入力操作でエミュレートするから、視差も無い極めて操作性の良い入力作業が実現される訳である。

【0037】また、上記実施例によれば、エミュレータ用に開発したソースプログラムがそのままハンディターミナル1用のソースファイルとなり、完全互換性を備えるので、プログラム開発が全て同一のマシン上で行うことが可能になる。この結果、従来必要とされていた実行ファイルE fのダウンロードDLと、これに応じてなされる実機デバッグDG 1とが省略されるから、極めて効率良いプログラム開発環境が実現できる。しかも、この場合、タッチパネル分割状態や、これに対応するキーコード割り当て状態が一目瞭然になる上、ハンディターミナル1のタッチパネル入力を開発マシン上で模倣できるから、デバッグ作業の効率も向上する。

【0038】さらに、ハンディターミナル用に作成したプログラムを顧客向けにデモンストレーションする場合、従来では、ハンディターミナル1そのものを使用して行っていた。この場合、表示面積が小さい液晶ディスプレイでその動作を表示するため、多人数へのデモンストレーションには不都合であった。しかしながら、これに替えて、上述した実施例のように、パーソナルコンピ

ュータでハンディターミナル1の動作をエミュレートすれば、大型のディスプレイによる多人数へのデモンストレーションが可能になる。

【0039】

【発明の効果】以上説明したように、この発明によれば、模倣手段は、一方のコンピュータ用に作成したプログラムで規定された入力形態を、他方のコンピュータが備える入力手段を用いて模倣するので、一方のコンピュータ側でなされるタッチパネル入力操作を、他方のコンピュータのマウス入力で模倣することができる。これにより、専用のタッチパネルを設ける必要がなくなるため、コスト的に有利となる。さらに、マウス入力でタッチパネル入力をエミュレートするようにしたから、視差による弊害もない。

【図面の簡単な説明】

【図1】この発明の一実施例を適用したプログラム開発手順の概要を示す図。

【図2】同実施例におけるエミュレータEMの機能構成を示す図。

【図3】同実施例により作成されたアプリケーションプログラムの概略動作を示すフローチャート。

【図4】同実施例により作成されたアプリケーションプログラムの概略動作を説明するための図。

【図5】同実施例におけるマウス入力ルーチンの動作を示すフローチャート。

【図6】同実施例における表示エリアE 1、E 2の表示例を示す図。

【図7】同実施例におけるマウス入力座標を説明するための図。

【図8】ハンディターミナル1の一構成例を示すブロック図。

【図9】従来のプログラム開発手順を説明するための図。

【符号の説明】

s f…ソースファイル、

c c…コンパイル、

o f…オブジェクトファイル、

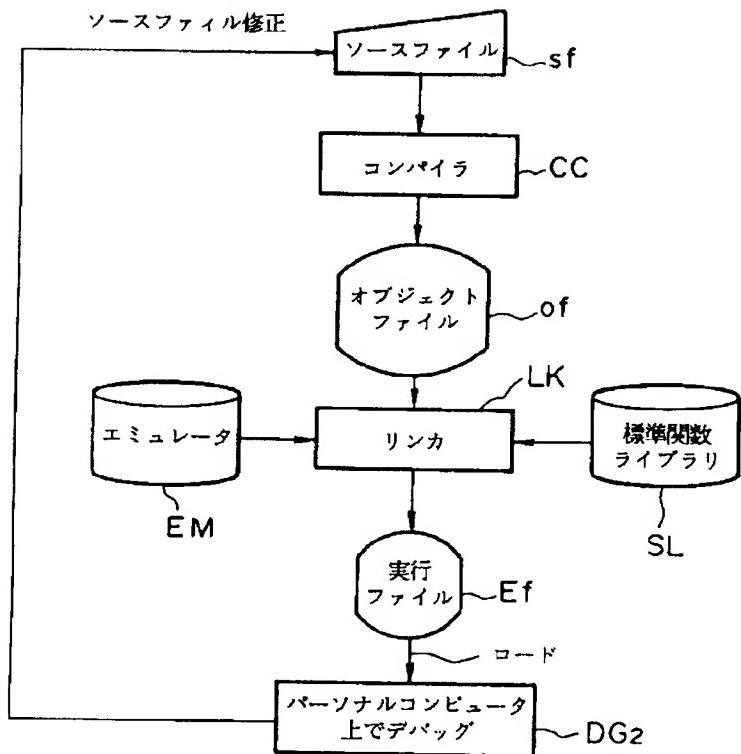
L K…リンク、

EM…エミュレータ（模倣手段）、

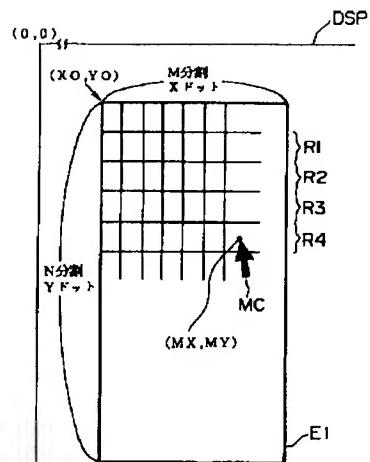
S L…標準関数ライブラリ、

E f…実行ファイル。

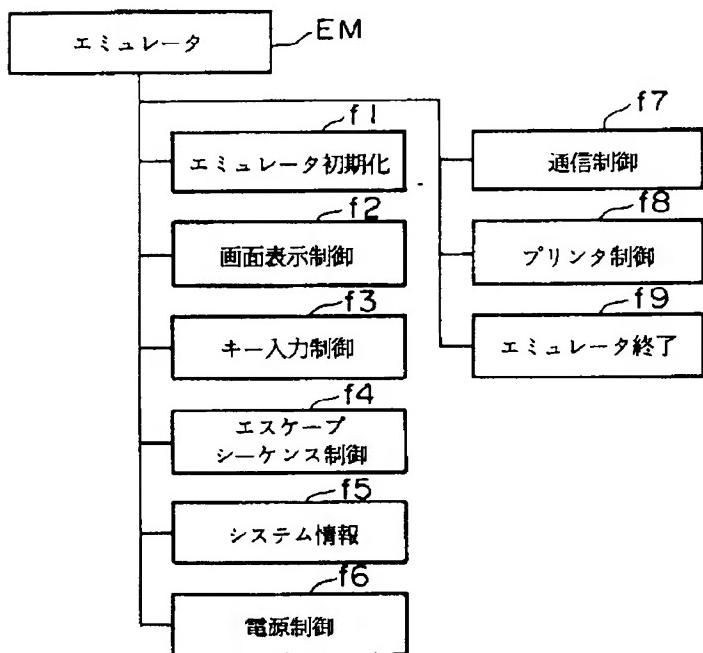
【図1】



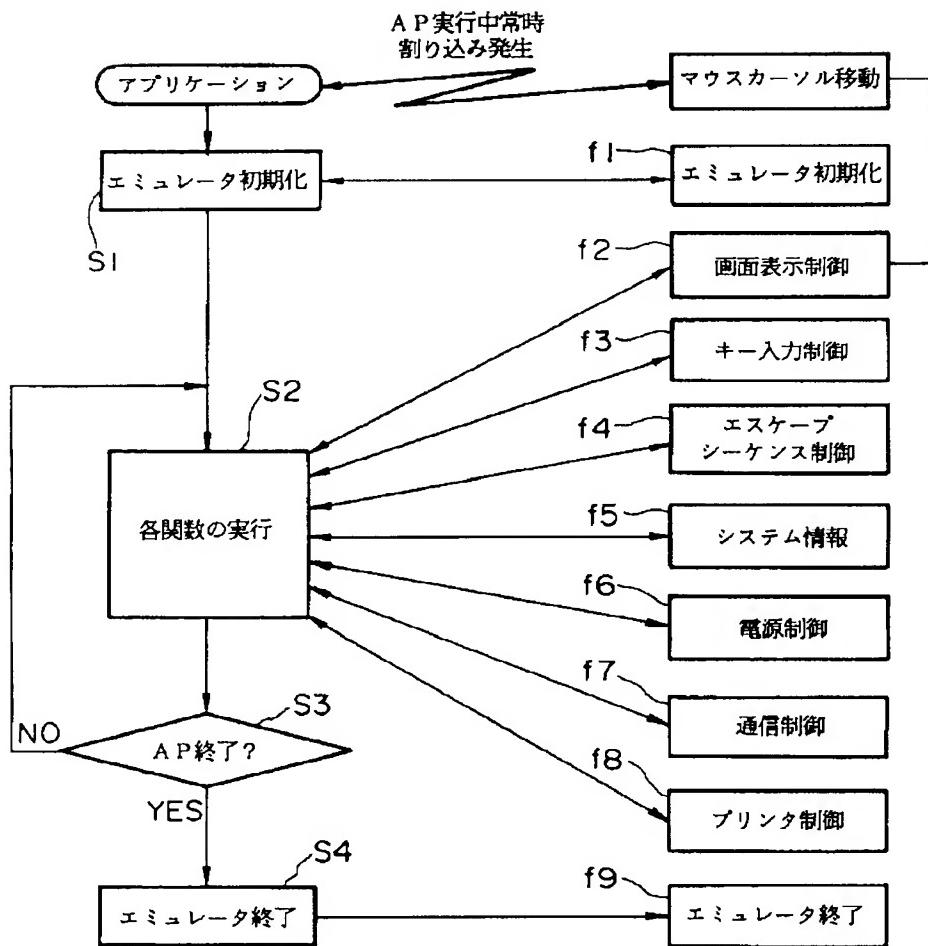
【図7】



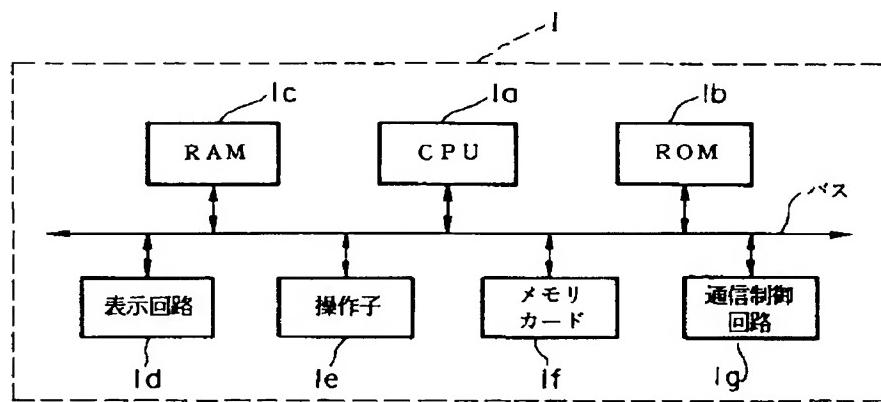
【図2】



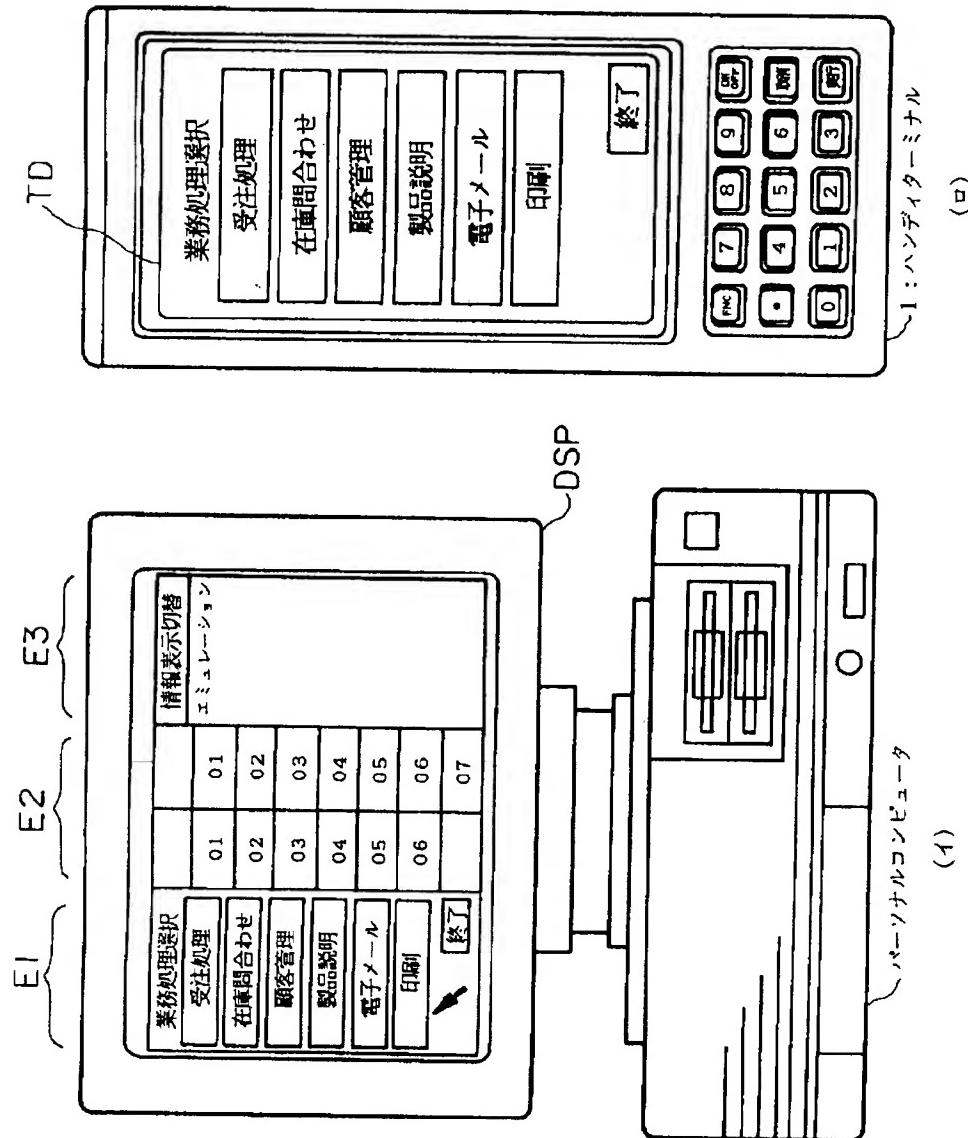
【図3】



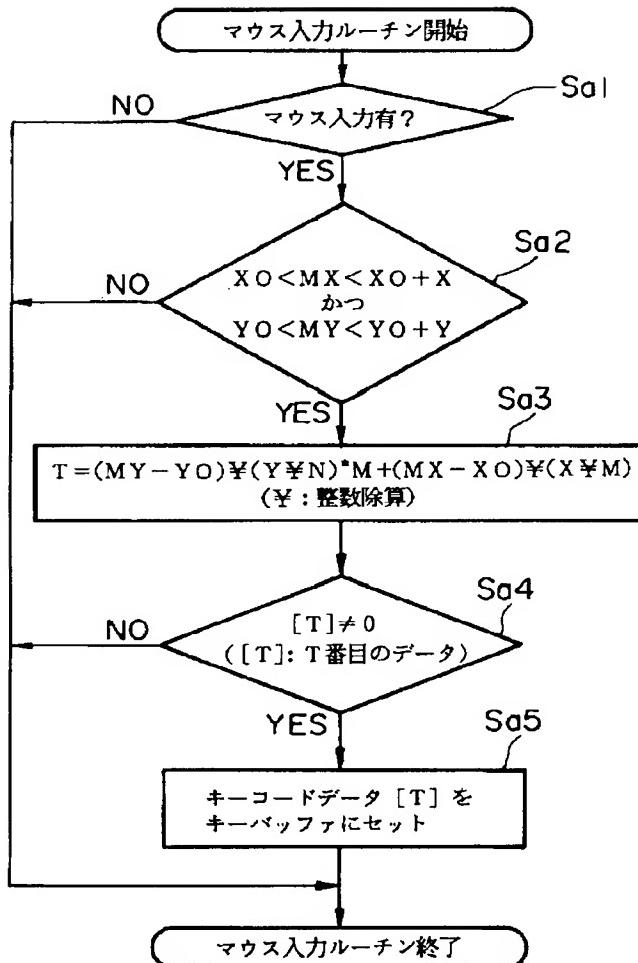
【図8】



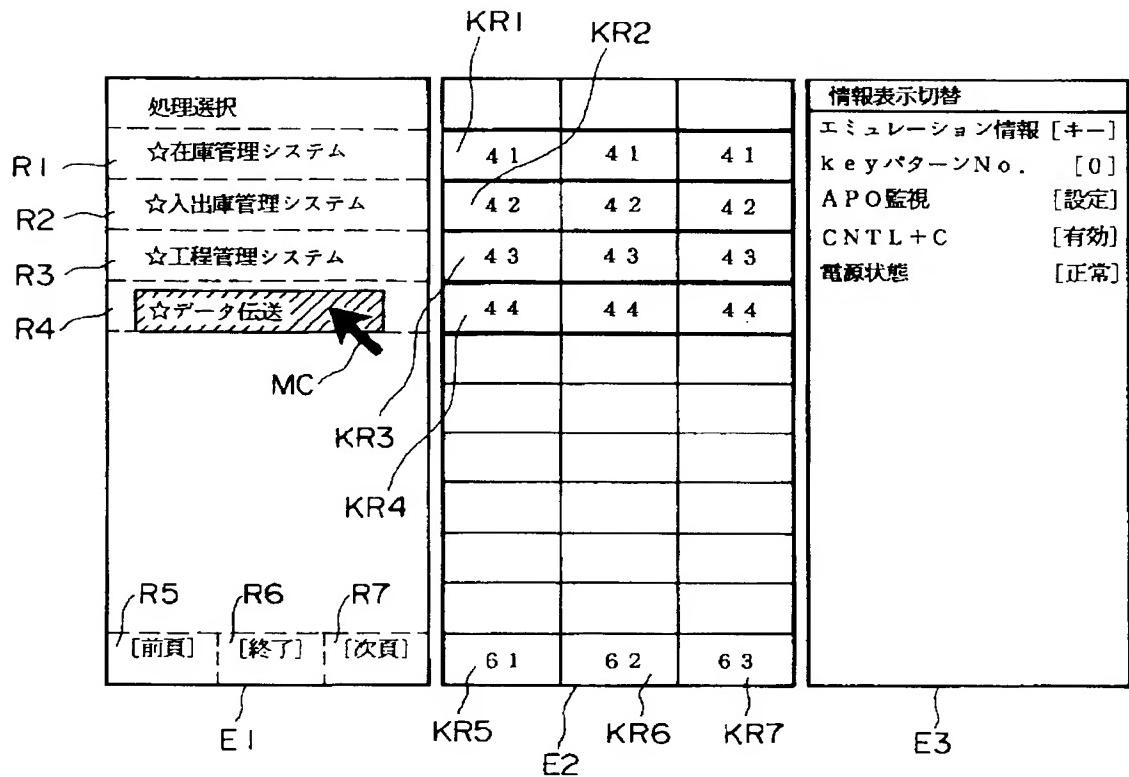
【図4】



【図5】



【図6】



[図9]

